Machine Learning Approaches to Link-Based Clustering

Zhongfei (Mark) Zhang[†], Bo Long[‡], Zhen Guo[†], Tianbing Xu[†], and Philip S. Yu[♮]

Abstract We have reviewed several state-of-the-art machine learning approaches to different types of link based clustering in this chapter. Specifically, we have presented the spectral clustering for heterogeneous relational data, the symmetric convex coding for homogeneous relational data, the citation model for clustering the special but popular homogeneous relational data – the textual documents with citations, the probabilistic clustering framework on mixed membership for general relational data, and the statistical graphical model for dynamic relational clustering. We have demonstrated the effectiveness of these machine learning approaches through empirical evaluations.

1 Introduction

Link information plays an important role in discovering knowledge from data. For link-based clustering, machine learning approaches provide pivotal strengths to develop effective solutions. In this chapter, we review several specific machine learning approaches to link-based clustering. We by no means mean that these approaches are exhaustive. Instead, our intention is to use these exemplar approaches to showcase the power of machine learning techniques to solve the general link-based clustering problem.

When we say link-based clustering, we mean the clustering of relational data. In other words, links are the relations among the data items or objects. Consequently,

Zhang, Guo, and Xu[†]
Computer Science Department, SUNY Binghamton, e-mail: \{zhongfei,zguo,txu\}@cs.binghamton.edu

Long[‡]
Yahoo! Labs, Yahoo! Inc., e-mail: blong@yahoo-inc.com

Yu[♮]
Dept. of Computer Science, Univ. of Illinois at Chicago, e-mail: psyu@cs.uic.edu

in the rest of this chapter, we use the terminologies of link-based clustering and relational clustering exchangeably. In general, relational data are those that have link information among the data items in addition to the classic attribute information for the data items. For relational data, we may categorize them in terms of the type of their relations into homogeneous relational data (relations exist among the same type of objects for all the data), heterogeneous relational data (relations only exist between data items of different types), general relational data (relations exist both among data items of the same type and between data items of different types), and dynamic relational data (there are time stamps for all the data items with relations to differentiate from all the previous types of relational data which are static). For all the specific machine learning approaches reviewed in this chapter, they are based on the mathematical foundations of matrix decomposition, optimization, and probability and statistics theory.

In this chapter, we review five different machine learning approaches tailored for different types of link-based clustering. Specifically, this chapter is organized as follows. In Section 2 we study the heterogeneous data clustering problem, and present an effective solution to this problem through spectral analysis. In Section 3 we study the homogeneous data clustering problem, and present an effective solution through symmetric convex coding. In Section 4, we study a special but very popular case of the homogeneous relational data, i.e., the data are the textual documents and the link information is the citation information, and present a generative citation model to capture the effective learning of the document topics. In Section 5, we study the general relational data clustering problem, and present a general probabilistic framework based on a generative model on mixed membership. In Section 6, we study the dynamic relational data clustering problem and present a solution to this problem under a statistical graphical model. Finally, we conclude this chapter in Section 8.

2 Heterogenous Relational Clustering through Spectral Analysis

Many real-world clustering problems involve data objects of multiple types that are related to each other, such as Web pages, search queries, and Web users in a Web search system, and papers, key words, authors, and conferences in a scientific publication domain. In such scenarios, using traditional methods to cluster each type of objects independently may not work well due to the following reasons.

First, to make use of relation information under the traditional clustering framework, the relation information needs to be transformed into features. In general, this transformation causes information loss and/or very high dimensional and sparse data. For example, if we represent the relations between Web pages and Web users as well as search queries as the features for the Web pages, this leads to a huge number of features with sparse values for each Web page. Second, traditional clustering approaches are unable to tackle with the interactions among the hidden structures of different types of objects, since they cluster data of single type based on static features. Note that the interactions could pass along the relations, i.e., there exists

influence propagation in multi-type relational data. Third, in some machine learning applications, users are not only interested in the hidden structure for each type of objects, but also the global structure involving multi-types of objects. For example, in document clustering, except for document clusters and word clusters, the relationship between document clusters and word clusters is also useful information. It is difficult to discover such global structures by clustering each type of objects individually.

Therefore, heterogeneous relational data have presented a great challenge for traditional clustering approaches. In this study [37], we present a general model, the collective factorization on related matrices, to discover the hidden structures of objects of different types based on both feature information and relation information. By clustering the objects of different types simultaneously, the model performs adaptive dimensionality reduction for each type of data. Through the related factorizations which share factors, the hidden structures of objects of different types may interact under the model. In addition to the cluster structures for each type of data, the model also provides information about the relation between clusters of objects of different types.

Under this model, we derive an iterative algorithm, the spectral relational clustering, to cluster the interrelated data objects of different types simultaneously. By iteratively embedding each type of data objects into low dimensional spaces, the algorithm benefits from the interactions among the hidden structures of data objects of different types. The algorithm has the simplicity of spectral clustering approaches but at the same time also is applicable to relational data with various structures. Theoretic analysis and experimental results demonstrate the promise and effectiveness of the algorithm. We also show that the existing spectral clustering algorithms can be considered as the special cases of the proposed model and algorithm. This provides a unified view to understanding the connections among these algorithms.

2.1 Model Formulation and Algorithm

In this section, we present a general model for clustering heterogeneous relational data in the spectral domain based on factorizing multiple related matrices.

Given m sets of data objects, $\mathcal{X}_1 = \{x_{11}, \ldots, x_{1n_1}\}, \ldots, \mathcal{X}_m = \{x_{m1}, \ldots, x_{mn_m}\}$, which refer to m different types of objects relating to each other, we are interested in simultaneously clustering $\mathcal{X}_1$ into k_1 disjoint clusters, ..., and $\mathcal{X}_m$ into k_m disjoint clusters. We call this task as *collective clustering on heterogeneous relational data*.

To derive a general model for collective clustering, we first formulate the Heterogeneous Relational Data (HRD) as a set of related matrices, in which two matrices are related in the sense that their row indices or column indices refer to the same set of objects. First, if there exist relations between $\mathcal{X}_i$ and $\mathcal{X}_j$ (denoted as $\mathcal{X}_i \sim \mathcal{X}_j$), we represent them as a relation matrix $R^{(ij)} \in \mathbb{R}^{n_i \times n_j}$, where an element $R^{(ij)}_{pq}$ denotes the relation between x_{ip} and x_{jq}. Second, a set of objects $\mathcal{X}_i$ may have its own features, which could be denoted by a feature matrix $F^{(i)} \in \mathbb{R}^{n_i \times f_i}$, where an element

$F_{pq}^{(i)}$ denotes the qth feature values for the object x_{ip} and f_i is the number of features for $\mathscr{X}_i$.

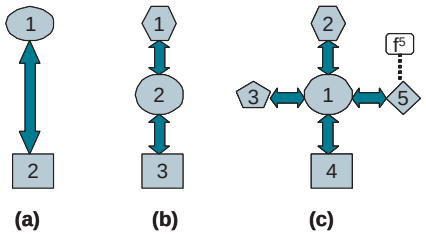


Fig. 1 Examples of the structures of the heterogeneous relational data.

Figure 1 shows three examples of the structures of HRD. Example (a) refers to a basic bi-type of relational data denoted by a relation matrix $R^{(12)}$, such as word-document data. Example (b) represents a tri-type of star-structured data, such as Web pages, Web users, and search queries in Web search systems, which are denoted by two relation matrices $R^{(12)}$ and $R^{(23)}$. Example (c) represents the data consisting of shops, customers, suppliers, shareholders, and advertisement media, in which customers (type 5) have features. The data are denoted by four relation matrices $R^{(12)}$, $R^{(13)}$, $R^{(14)}$ and $R^{(15)}$, and one feature matrix $F^{(5)}$.

It has been shown that the hidden structure of a data matrix can be explored by its factorization [14,40]. Motivated by this observation, we propose a general model for collective clustering, which is based on factorizing the multiple related matrices. In HRD, the cluster structure for a type of objects $\mathscr{X}_i$ may be embedded in multiple related matrices; hence it can be exploited in multiple related factorizations. First, if $\mathscr{X}_i \sim \mathscr{X}_j$, then the cluster structures of both $\mathscr{X}_i$ and $\mathscr{X}_j$ are reflected in the triple factorization of their relation matrix $R^{(ij)}$ such that $R^{(ij)} \approx C^{(i)}A^{(ij)}(C^{(j)})^T$ [40], where $C^{(i)} \in \{0,1\}^{n_i \times k_i}$ is a *cluster indicator matrix* for $\mathscr{X}_i$ such that $\sum_{q=1}^{k_i} C_{pq}^{(i)} = 1$ and $C_{pq}^{(i)} = 1$ denotes that the pth object in $\mathscr{X}_i$ is associated with the qth cluster. Similarly $C^{(j)} \in \{0,1\}^{n_j \times k_j}$. $A^{(ij)} \in \mathbb{R}^{k_i \times k_j}$ is the *cluster association matrix* such that A_{pq}^{ij} denotes the association between cluster p of $\mathscr{X}_i$ and cluster q of $\mathscr{X}_j$. Second, if $\mathscr{X}_i$ has a feature matrix $F^{(i)} \in \mathbb{R}^{n_i \times f_i}$, the cluster structure is reflected in the factorization of $F^{(i)}$ such that $F^{(i)} \approx C^{(i)}B^{(i)}$, where $C^{(i)} \in \{0,1\}^{n_i \times k_i}$ is a cluster indicator matrix, and $B^{(i)} \in \mathbb{R}^{k_i \times f_i}$ is the feature basis matrix which consists of k_i basis (cluster center) vectors in the feature space.

Based on the above discussions, formally we formulate the task of collective clustering on HRD as the following optimization problem. Considering the most general case, we assume that in HRD, every pair of $\mathscr{X}_i$ and $\mathscr{X}_j$ is related to each other and every $\mathscr{X}_i$ has a feature matrix $F^{(i)}$.

Definition 1. Given m positive numbers $\{k_i\}_{1 \le i \le m}$ and HRD $\{\mathscr{X}_1, \ldots, \mathscr{X}_m\}$, which is described by a set of relation matrices $\{R^{(ij)} \in \mathbb{R}^{n_i \times n_j}\}_{1 \le i < j \le m}$, a set of feature matrices $\{F^{(i)} \in \mathbb{R}^{n_i \times f_i}\}_{1 \le i \le m}$, as well as a set of weights $w_a^{(ij)}, w_b^{(i)} \in R_+$ for differ-

ent types of relations and features, the task of the collective clustering on the HRD is to minimize

$$L = \sum_{1 \le i < j \le m} w_a^{(ij)} ||R^{(ij)} - C^{(i)} A^{(ij)} (C^{(j)})^T||^2 + \sum_{1 \le i \le m} w_b^{(i)} ||F^{(i)} - C^{(i)} B^{(i)}||^2 \quad (1)$$

w.r.t. $C^{(i)} \in \{0,1\}^{n_i \times k_i}$, $A^{(ij)} \in \mathbb{R}^{k_i \times k_j}$, and $B^{(i)} \in \mathbb{R}^{k_i \times f_i}$ subject to the constraints: $\sum_{q=1}^{k_i} C_{pq}^{(i)} = 1$, where $1 \le p \le n_i$, $1 \le i < j \le m$, and $||\cdot||$ denotes the Frobenius norm for a matrix.

We call the model proposed in Definition 1 as the Collective Factorization on Related Matrices (CFRM).

The CFRM model clusters heterogeneously interrelated data objects simultaneously based on both relation and feature information. The model exploits the interactions between the hidden structures of different types of objects through the related factorizations which share matrix factors, i.e., cluster indicator matrices. Hence, the interactions between hidden structures work in two ways. First, if $\mathscr{X}_i \sim \mathscr{X}_j$, the interactions are reflected as the duality of row clustering and column clustering in $R^{(ij)}$. Second, if two types of objects are indirectly related, the interactions pass along the relation "chains" by a chain of related factorizations, i.e., the model is capable of dealing with influence propagation. In addition to local cluster structure for each type of objects, the model also provides the global structure information by the cluster association matrices, which represent the relations among the clusters of different types of objects.

Based on the CFRM model, we derive an iterative algorithm, called Spectral Relational Clustering (SRC) algorithm [37]. The specific derivation of the algorithm and the proof of the convergence of the algorithm refer to [37]. Further, In Long et al [37], it is shown that the CFRM model as well as the SRC algorithm is able to handle the general case of heterogeneous relational data, and many existing methods in the literature are either the special cases or variations of this model. Specifically, it is shown that the classic k-means [53], the spectral clustering methods based on graph partitioning [42,43,43], and the Bipartite Spectral Graph Partitioning (BSGP) [18,52] are all the special cases of this general model.

2.2 Experiments

The SRC algorithm is evaluated on two types of HRD, bi-type relational data and tri-type star-structured data as shown in Figure 1(a) and Figure 1(b), which represent two basic structures of HRD and arise frequently in real applications.

The datasets used in the experiments are mainly based on the 20-Newsgroups data [34] which contain about 20,000 articles from 20 newsgroup. We pre-process the data by removing stop words and file headers and selecting top 2000 words

by the mutual information. The word-document matrix R is based on *tf.idf* and each document vector is normalized to the unit norm vector. In the experiments the classis k-means is used for initialization and the final performance score for each algorithm is the average of the 20 test runs unless stated otherwise.

2.2.1 Clustering on Bi-type Relational Data

In this section we report experiments on bi-type relational data, word-document data, to demonstrate the effectiveness of SRC as a novel co-clustering algorithm. A representative spectral clustering algorithm, Normalized-Cut (NC) spectral clustering [42, 43], and BSGP [18], are used for comparisons.

The graph affinity matrix for NC is R^TR, i.e., the cosine similarity matrix. In NC and SRC, the leading k eigenvectors are used to extract the cluster structure, where k is the number of document clusters. For BSGP, the second to the $(\lceil \log_2 k \rceil + 1)$th leading singular vectors are used [18]. K-means is adopted to postprocess the eigenvectors. Before postprocessing, the eigenvectors from NC and SRC are normalized to the unit norm vector and the eigenvectors from BSGP are normalized as described by [18]. Since all the algorithms have random components resulting from k-means or itself, at each test we conduct three trials with random initializations for each algorithm and the optimal one provides the performance score for that test run. To evaluate the quality of document clusters, we elect to use the Normalized Mutual Information (NMI) [44], which is a standard measure for the clustering quality.

At each test run, five datasets, multi2 (NG 10, 11), multi3(NG 1,10,20), multi5 (NG 3, 6, 9, 12, 15), multi8 (NG 3, 6, 7, 9, 12, 15, 18, 20) and multi10 (NG 2, 4, 6, 8, 10, 12 ,14 ,16 ,18,20), are generated by randomly sampling 100 documents from each newsgroup. Here NG i means the ith newsgroup in the original order. For the numbers of document clusters, we use the numbers of the true document classes. For the numbers of word clusters, there are no options for BSGP, since they are restricted to equal to the numbers of document clusters. For SRC, it is flexible to use any number of word clusters. Since how to choose the optimal number of word clusters is beyond the scope of this study, we simply choose one more word cluster than the corresponding document clusters, i.e., 3,4, 6, 9, and 11. This may not be the best choice but it is good enough to demonstrate the flexibility and effectiveness of SRC.

In Figure 2, (a), (b) and (c) show three document embeddings of a multi2 sample, which is sampled from two close newsgroups, *rec.sports.baseball* and *rec.sports.hockey*. In this example, when NC and BSGP fail to separate the document classes, SRC still provides a satisfactory separation. The possible explanation is that the adaptive interactions among the hidden structures of word clusters and document clusters remove the noise to lead to better embeddings. (d) shows a typical run of the SRC algorithm.

Table 1 shows NMI scores on all the datasets. We observe that SRC performs better than NC and BSGP on all data sets. This verifies the hypothesis that benefiting from the interactions of the hidden structures of objects with different types, the

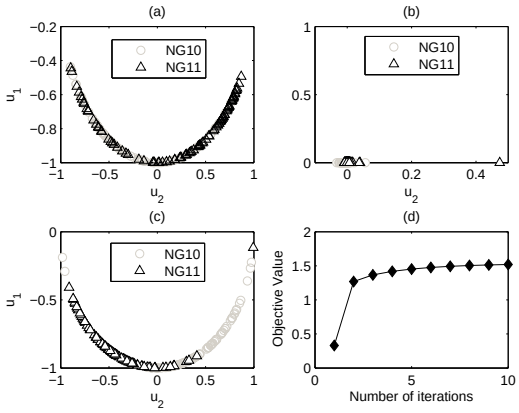


Fig. 2 (a), (b), and (c) are document embeddings of multi2 dataset produced by NC, BSGP, and SRC, respectively (u_1 and u_2 denote first and second eigenvectors, respectively). (d) is an iteration curve for SRC.

Table 1 NMI comparisons of SRC, NC, and BSGP algorithms

Data set	SRC	NC	BSGP
multi2	0.4979	0.1036	0.1500
multi3	0.5763	0.4314	0.4897
multi5	0.7242	0.6706	0.6118
multi8	0.6958	0.6192	0.5096
multi10	0.7158	0.6292	0.5071

SRC's adaptive dimensionality reduction has advantages over the dimensionality reduction of the existing spectral clustering algorithms.

2.2.2 Clustering on Tri-type Relational Data

In this section, we report the experiments on tri-type star-structured relational data to evaluate the effectiveness of SRC in comparison with other two algorithms for HRD clustering. One is based on the *m*-partite graph partitioning, Consistent Bipartite Graph Co-partitioning (CBGC) [24] (we thank the authors for providing the executable program of CBGC). The other is Mutual Reinforcement K-means (MRK), which is implemented based on the idea of mutual reinforcement clustering.

The first dataset is synthetic data, in which two relation matrices, $R^{(12)}$ with 80-by-100 dimension and $R^{(23)}$ with 100-by-80 dimension, are binary matrices with 2-by-2 block structures. $R^{(12)}$ is generated based on the block structure $\left[\begin{smallmatrix} 0.9 & 0.7 \\ 0.8 & 0.9 \end{smallmatrix}\right]$, i.e., the objects in cluster 1 of $\mathscr{X}^{(1)}$ is related to the objects in cluster 1 of $\mathscr{X}^{(2)}$ with probability 0.9, and so on so forth. $R^{(23)}$ is generated based on the block structure $\left[\begin{smallmatrix} 0.6 & 0.7 \\ 0.7 & 0.6 \end{smallmatrix}\right]$. Each type of objects has two equal size clusters. It is not a trivial task to

Table 2 Taxonomy structures for three datasets

DATA SET	TAXONOMY STRUCTURE
TM1	{NG10, NG11}, {NG17, NG18, NG19}
TM2	{NG2, NG3}, {NG8, NG9}, {NG12, NG13}
TM3	{NG4, NG5}, {NG8, NG9}, {NG14, NG15}, {NG17, NG18}

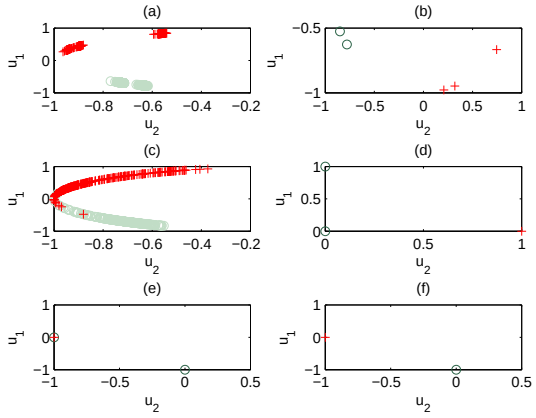


Fig. 3 Three pairs of embeddings of documents and categories for the TM1 dataset produced by SRC with different weights: (a) and (b) with $w_a^{(12)} = 1, w_a^{(23)} = 1$; (c) and (d) with $w_a^{(12)} = 1, w_a^{(23)} = 0$; (e) and (f) with $w_a^{(12)} = 0, w_a^{(23)} = 1$.

identify the cluster structure of this dataset, since the block structures are subtle. We denote this dataset as Binary Relation Matrices (TRM) data.

Other three datasets are built based on the 20-newsgroups data for hierarchical taxonomy mining and document clustering. In the field of text categorization, hierarchical taxonomy classification is widely used to obtain a better trade-off between effectiveness and efficiency than flat taxonomy classification. To take advantage of hierarchical classification, one must mine a hierarchical taxonomy from the dataset. We can see that words, documents, and categories formulate tri-type relational data, which consist of two relation matrices, a word-document matrix $R^{(12)}$ and a document-category matrix $R^{(23)}$ [24].

The true taxonomy structures for the three datasets, TM1, TM2, and TM3, are listed in Table 2. For example, TM1 dataset is sampled from five categories, in which NG10 and NG11 belong to the same high level category *res.sports* and NG17, NG18, and NG19 belong to the same high level category *talk.politics*. Therefore, for the TM1 dataset, the expected clustering result on categories should be {NG10, NG11} and {NG17, NG18, NG19} and the documents should be clustered into two clusters according to their categories. The documents in each dataset are generated by sampling 100 documents from each category.

The number of the clusters used for documents and categories are 2, 3, and 4 for TM1, TM2, and TM3, respectively. For the number of word clusters, we adopt the number of categories, i.e., 5, 6 and 8. For the weights $w_a^{(12)}$ and $w_a^{(23)}$, we simply use equal weight, i.e., $w_a^{(12)} = w_a^{(23)} = 1$. Figure 3 illustrates the effects of different weights on embeddings of documents and categories. When $w_a^{(12)} = w_a^{(23)} = 1$, i.e., SRC makes use of both word-document relations and document-category relations, both documents and categories are separated into two clusters very well as in (a) and (b) of Figure 3, respectively; when SRC makes use of only the word-document relations, the documents are separated with partial overlapping as in (c) and the categories are randomly mapped to a couple of points as in (d); when SRC makes use of only the document-category relations, both documents and categories are incorrectly overlapped as in (e) and (f), respectively, since the document-category matrix itself does not provide any useful information for the taxonomy structure.

The performance comparison is based on the cluster quality of documents, since the better it is, the more accurate we can identify the taxonomy structures. Table 3 shows NMI comparisons of the three algorithms on the four datasets. The NMI score of CBGC is available only for BRM dataset because the CBGC program provided by the authors only works for the case of two clusters and small size matrices. We observe that SRC performs better than MRK and CBGC on all datasets. The comparison shows that among the limited efforts in the literature attempting to cluster multi-type interrelated objects simultaneously, SRC is an effective one to identify the cluster structures of HRD.

Table 3 NMI comparisons of SRC, MRK, and CBGC algorithms

Data set	SRC	MRK	CBGC
BRM	0.6718	0.6470	0.4694
TM1	1	0.5243	–
TM2	0.7179	0.6277	–
TM3	0.6505	0.5719	–

3 Homogeneous Relational Clustering through Convex Coding

The most popular way to solve the problem of clustering the homogeneous relational data such as similarity-based relational data is to formulate it as a graph partitioning problem, which has been studied for decades. Graph partitioning seeks to cut a given graph into disjoint subgraphs which correspond to disjoint clusters based on a certain edge cut objective. Recently, graph partitioning with an edge cut objective has been shown to be mathematically equivalent to an appropriate weighted kernel k-means objective function [16, 17]. The assumption behind the graph partitioning formulation is that since the nodes within a cluster are similar to each other,

they form a dense subgraph. However, in general this is not true for relational data, i.e., the clusters in relational data are not necessarily *dense* clusters consisting of strongly-related objects.

Figure 4 shows the relational data of four clusters, which are of two different types. In Figure 4, $\mathscr{C}_1 = \{v_1, v_2, v_3, v_4\}$ and $\mathscr{C}_2 = \{v_5, v_6, v_7, v_8\}$ are two traditional dense clusters within which objects are strongly related to each other. However, $\mathscr{C}_3 = \{v_9, v_{10}, v_{11}, v_{12}\}$ and $\mathscr{C}_4 = \{v_{13}, v_{14}, v_{15}, v_{16}\}$ also form two *sparse* clusters, within which the objects are not related to each other, but they are still "similar" to each other in the sense that they are related to the same set of other nodes. In Web mining, this type of cluster could be a group of music "fans" Web pages which share the same taste on the music and are linked to the same set of music Web pages but are not linked to each other [33]. Due to the importance of identifying this type of clusters (communities), it has been listed as one of the five algorithmic challenges in Web search engines [28]. Note that the cluster structure of the relation data in Figure 4 cannot be correctly identified by graph partitioning approaches, since they look for only dense clusters of strongly related objects by cutting the given graph into subgraphs; similarly, the pure bi-partite graph models cannot correctly identify this type of cluster structures. Note that re-defining the relations between the objects does not solve the problem in this situation, since there exist both dense and sparse clusters.

If the homogeneous relational data are dissimilarity-based, to apply graph partitioning approaches to them, we need extra efforts on appropriately transforming them into similarity-based data and ensuring that the transformation does not change the cluster structures in the data. Hence, it is desirable for an algorithm to be able to identify the cluster structures no matter which type of relational data is given. This is even more desirable in the situation where the background knowledge about the meaning of the relations is not available, i.e., we are given only a relation matrix and do not know if the relations are similarities or dissimilarities.

In this section, we present a general model for relational clustering based on symmetric convex coding of the relation matrix [36]. The model is applicable to the general homogeneous relational data consisting of only pairwise relations typically without other knowledge; it is capable of learning both dense and sparse clusters at the same time; it unifies the existing graph partition models to provide a generalized theoretical foundation for relational clustering. Under this model, we derive iterative bound optimization algorithms to solve the symmetric convex coding for two important distance functions, Euclidean distance and generalized I-divergence. The algorithms are applicable to general relational data and at the same time they can be easily adapted to learn a specific type of cluster structure. For example, when applied to learning only dense clusters, they provide new efficient algorithms for graph partitioning. The convergence of the algorithms is theoretically guaranteed. Experimental evaluation and theoretical analysis show the effectiveness and great potential of the proposed model and algorithms.

3.1 Model Formulation and Algorithms

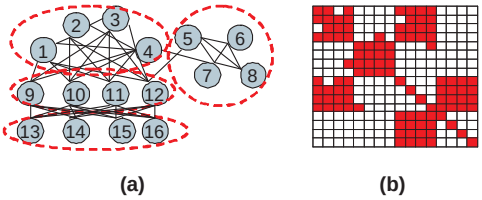


Fig. 4 The graph (a) and relation matrix (b) of the relational data with different types of clusters. In (b), the dark color denotes 1 and the light color denotes 0.

In this section, we describe a general model for homogeneous relational clustering. Let us first consider the relational data in Figure 4. An interesting observation is that although the different types of clusters look so different in the graph from Figure 4(a), they all demonstrate block patterns in the relation matrix of Figure 4(b) (without loss of generality, we arrange the objects from the same cluster together to make the block patterns explicit). Motivated by this observation, we propose the Symmetric Convex Coding (SCC) model to cluster relational data by learning the block pattern of a relation matrix. Since in most applications, the relations are of non-negative values and undirected, homogeneous relational data can be represented as non-negative, symmetric matrices. Therefore, the definition of SCC is given as follows.

Definition 2. Given a symmetric matrix $A \in \mathbb{R}_+$, a distance function $\mathfrak{D}$ and a positive number k, the symmetric convex coding is given by the minimization,

$$\min_{\substack{C \in \mathbb{R}_+^{n \times k}, B \in \mathbb{R}_+^{k \times k} \\ C\mathbf{1}=\mathbf{1}}} \mathfrak{D}(A, CBC^T). \tag{2}$$

According to Definition 2, the elements of C are between 0 and 1 and the sum of the elements in each row of C equal to 1. Therefore, SCC seeks to use the convex combination of the *prototype matrix* B to approximate the original relation matrix. The factors from SCC have intuitive interpretations. The factor C is the soft membership matrix such that C_{ij} denotes the weight that the ith object associates with the jth cluster. The factor B is the prototype matrix such that B_{ii} denotes the connectivity within the ith cluster and B_{ij} denotes the connectivity between the ith cluster and the jth cluster.

SCC provides a general model to learn various cluster structures from relational data. Graph partitioning, which focuses on learning dense cluster structure, can be formulated as a special case of the SCC model. We propose the following theorem to show that the various graph partitioning objective functions are mathematically equivalent to a special case of the SCC model. Since most graph partitioning objective functions are based on the hard cluster membership, in the following theorem

we change the constraints on C as $C \in \mathbb{R}_+$ and $C^TC = I_k$ to make C to be the following cluster indicator matrix,

$$C_{ij} = \begin{cases} \frac{1}{|\pi_j|^{\frac{1}{2}}} & \text{if } v_i \in \pi_j \\ 0 & \text{otherwise} \end{cases}$$

where $|\pi_j|$ denotes the number of nodes in the jth cluster.

Theorem 1. *The hard version of SCC model under Euclidean distance function and $B = rI_k$ for $r > 0$, i.e.,*

$$\min_{\substack{C \in \mathbb{R}_+^{n\times k}, B \in \mathbb{R}_+^{k\times k} \\ C^TC = I_k}} ||A - C(rI_k)C^T||^2 \tag{3}$$

is equivalent to the maximization

$$\max tr(C^TAC), \tag{4}$$

where tr denots the trace of a matrix.

The proof of Theorem 1 may be found in [36].

Theorem 1 states that with the prototype matrix B restricted to be of the form rI_k, SCC under Euclidean distance is reduced to the trace maximization in (4). Since various graph partitioning objectives, such as ratio association [43], normalized cut [43], ratio cut [9], and Kernighan-Lin objective [32], can be formulated as the trace maximization [16, 17], Theorem 1 establishes the connection between the SCC model and the existing graph partitioning objective functions. Based on this connection, it is clear that the existing graph partitioning models make an implicit assumption for the cluster structure of the relational data, i.e., the clusters are not related to each other (the off-diagonal elements of B are zeroes) and the nodes within clusters are related to each other in the same way (the diagonal elements of B are r). This assumption is consistent with the intuition about the graph partitioning, which seeks to "cut" the graph into k separate subgraphs corresponding to the strongly-related clusters.

With Theorem 1 we may put other types of structural constraints on B to derive new graph partitioning models. For example, we fix B as a general diagonal matrix instead of rI_k, i.e., the model fixes the off-diagonal elements of B as zero and learns the diagonal elements of B. This is a more flexible graph partitioning model, since it allows the connectivity within different clusters to be different. More generally, we can use B to restrict the model to learn other types of the cluster structures. For example, by fixing diagonal elements of B as zeros, the model focuses on learning only spare clusters (corresponding to bi-partite or k-partite subgraphs), which are important for Web community learning [28, 33]. In summary, the prototype matrix B not only provides the intuition for the cluster structure of the data, but also provides a simple way to adapt the model to learn specific types of cluster structures.

Now efficient algorithms for the SCC model may be derived under two popular distance functions, Euclidean distance and generalized I-divergence. SCC under the Euclidean distance, i.e., an algorithm alternatively updating B and C until convergence, is derived and called SCC-ED [36].

If the task is to learn the dense clusters from similarity-based relational data as the graph partitioning does, SCC-ED can achieve this task simply by fixing B as the identity matrix and updating only C until convergence. In other words, these updating rules provide a new and efficient graph partitioning algorithm, which is computationally more efficient than the popular spectral graph partitioning approaches which involve expensive eigenvector computation (typically $O(n^3)$) and the extra post-processing [51] on eigenvectors to obtain the clustering. Compared with the multi-level approaches such as METIS [31], this new algorithm does not restrict clusters to have an equal size.

Another advantage of the SCC-ED algorithm is that it is very easy for the algorithm to incorporate constraints on B to learn a specific type of cluster structures. For example, if the task is to learn the sparse clusters by constraining the diagonal elements of B to be zero, we can enforce this constraint simply by initializing the diagonal elements of B as zeros. Then, the algorithm automatically only updates the off-diagonal elements of B and the diagonal elements of B are 'locked' to zeros.

Yet another interesting observation about SCC-ED is that if we set $\alpha = 0$ to change the updating rule for C into the following,

$$C = \tilde{C} \odot \left(\frac{A\tilde{C}B}{\tilde{C}B\tilde{C}^T\tilde{C}B}\right)^{\frac{1}{4}}, \tag{5}$$

the algorithm actually provides the symmetric conic coding. This has been touched in the literature as the symmetric case of non-negative factorizaion [8,19,40]. Therefore, SCC-ED under $\alpha = 0$ also provides a theoretically sound solution to the symmetric nonnegative matrix factorization.

Under the generalized I-divergence, the SCC objective function is given as follows,

$$D(A||CBC^T) = \sum_{ij}\left(A_{ij}\log\frac{A_{ij}}{[CBC^T]_{ij}} - A_{ij} + [CBC^T]_{ij}\right) \tag{6}$$

Similarly, an alternative bound optimization algorithm is derived for this objective function, called SCC-GI [36], which provides another new relational clustering algorithm. Again, when applied to the similarity-based relational data of dense clusters, SCC-GI provides another new and efficient graph partitioning algorithm.

The specific derivation of the two algorithms refers to [36], where the complexity and the convergence issues of the algorithms are discussed.

Table 4 Summary of the synthetic relational data

Graph	Parameter	n	k
syn1	$\begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$	900	3
syn2	$1 - syn1$	900	3
syn3	$\begin{bmatrix} 0 & 0.1 & 0.1 \\ 0.1 & 0 & 0.2 \\ 0.1 & 0.2 & 0 \end{bmatrix}$	900	3
syn4	$[0,1]^{10\times 10}$	5000	10

3.2 Experiments

This section provides empirical evidence to show the effectiveness of the SCC model and algorithms in comparison with two representative graph partitioning algorithms, a spectral approach, Normalized Cut (NC) [43], and a multilevel algorithm, METIS [31].

3.2.1 Datasets and Parameter Setting

The datasets used in the experiments include synthetic datasets with various cluster structures and real datasets based on various text data from the 20-newsgroups [34], WebACE, and TREC [30].

First, we use synthetic binary relational data to simulate homegeneous relational data with different types of clusters such as dense clusters, sparse clusters, and mixed clusters. All the synthetic relational data are generated based on Bernoulli distribution. The distribution parameters to generate the graphs are listed in the second column of Table 4 as matrices (true prototype matrices for the data). In a parameter matrix P, P_{ij} denotes the probability that the nodes in the ith cluster are connected to the nodes in the jth cluster. For example, in dataset syn3, the nodes in cluster 2 are connected to the nodes in cluster 3 with probability 0.2 and the nodes within a cluster are connected to each other with probability 0. Syn2 is generated by using 1 minus syn1. Hence, syn1 and syn2 can be viewed as a pair of similarity/dissimilarity data. Dataset syn4 has ten clusters mixing with dense clusters and sparse clusters. Due to the space limit, its distribution parameters are omitted here. Totally syn4 has 5000 nodes and about 2.1 million edges.

The graphs based on the text data have been widely used to test graph partitioning algorithms [18,20,52]. Note that there also exist feature-based algorithms to directly cluster documents based on word features. However, in this study our focus is on the clustering based on relations instead of features. Hence graph clustering algorithms are used in comparisons. We use various datasets from the 20-newsgroups [34], WebACE, and TREC [30], which cover datasets of different sizes, different balances, and different levels of difficulties. We construct relational data for each text dataset such that objects (documents) are related to each other with cosine similarities between the term-frequency vectors. A summary of all the datasets to construct

relational data used in this study is shown in Table 5, in which *n* denotes the number of objects in the relational data, *k* denotes the number of true clusters, and *balance* denotes the size ratio of the smallest clusters to the largest clusters.

Table 5 Summary of relational data based on text data sets.

Name	n	k	Balance	Source
tr11	414	9	0.046	TREC
tr23	204	6	0.066	TREC
NG17-19	1600	3	0.5	20-newsgroups
NG1-20	14000	20	1.0	20-newsgroups
k1b	2340	6	0.043	WebACE
hitech	2301	6	0.192	TREC
classic3	3893	3	0.708	MEDLINE/ CISI/CRANFILD

For the number of clusters *k*, we simply use the number of the true clusters. Note that how to choose the optimal number of clusters is a nontrivial model selection problem and beyond the scope of this study. For performance measure, we elect to use the Normalized Mutual Information (NMI) [44] between the resulting cluster labels and the true cluster labels, which is a standard measure for the clustering quality. The final performance score is the average of ten runs.

3.2.2 Results and Discussion

Table 6 NMI comparisons of NC, METIS, SCC-ED, and SCC-GI algorithms

Data	NC	METIS	SCC-ED	SCC-GI
syn1	0.9652 ± 0.031	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
syn2	0.8062 ± 0.52	0.000 ± 0.00	0.9038 ± 0.045	$\mathbf{0.9753 \pm 0.011}$
syn3	0.636 ± 0.152	0.115 ± 0.001	0.915 ± 0.145	$\mathbf{1.000 \pm 0.000}$
syn4	0.611 ± 0.032	0.638 ± 0.001	0.711 ± 0.043	$\mathbf{0.788 \pm 0.041}$
tr11	0.629 ± 0.039	0.557 ± 0.001	0.6391 ± 0.033	$\mathbf{0.661 \pm 0.019}$
tr23	0.276 ± 0.023	0.138 ± 0.004	$\mathbf{0.335 \pm 0.043}$	0.312 ± 0.099
NG17-19	0.002 ± 0.002	0.091 ± 0.004	0.1752 ± 0.156	$\mathbf{0.225 \pm 0.045}$
NG1-20	0.510 ± 0.004	$\mathbf{0.526 \pm 0.001}$	0.5041 ± 0.156	0.519 ± 0.010
k1b	0.546 ± 0.021	0.243 ± 0.000	0.537 ± 0.023	$\mathbf{0.591 \pm 0.022}$
hitech	0.302 ± 0.005	$\mathbf{0.322 \pm 0.001}$	0.319 ± 0.012	0.319 ± 0.018
classic3	0.621 ± 0.029	0.358 ± 0.000	0.642 ± 0.043	$\mathbf{0.822 \pm 0.059}$

Table 6 shows the NMI scores of the four algorithms on synthetic and real relational data. Each NMI score is the average of ten test runs and the standard deviation is also reported. We observe that although there is no single winner on all the data, for most data SCC algorithms perform better than or close to NC and METIS. Especially, SCC-GI provides the best performance on eight of the eleven datasets.

For the synthetic dataset syn1, almost all the algorithms provide perfect NMI score, since the data are generated with very clear dense cluster structures, which can be seen from the parameter matrix in Table 4. For dataset syn2, the dissimilarity version of syn1, we use exactly the same set of true cluster labels as that of syn1 to measure the cluster quality; the SCC algorithms still provide almost perfect NMI score; however, METIS totally fails on syn2, since in syn2 the clusters have the form of sparse clusters, and based on the edge cut objective, METIS looks for only dense clusters. An interesting observation is that the NC algorithm does not totally fail on syn2 and in fact it provides a satisfactory NMI score. This is due to that although the original objective of the NC algorithm focuses on dense clusters (its objective function can be formulated as the trace maximization in (4)), after relaxing C to an arbitrary orthonormal matrix, what NC actually does is to embed cluster structures into the eigen-space and to discover them by post-processing the eigenvectors. Besides the dense cluster structures, sparse cluster structures could also have a good embedding in the eigen-space under a certain condition.

In dataset syn3, the relations within clusters are sparser than the relations between clusters, i.e., it also has sparse clusters, but the structure is more subtle than syn2. We observe that NC does not provide a satisfactory performance and METIS totally fails; in the mean time, SCC algorithms identify the cluster structure in syn3 very well. Dataset syn4 is a large relational dataset of ten clusters consisting of four dense clusters and six sparse clusters; we observe that the SCC algorithms perform significantly better than NC and METIS on it, since they can identify both dense clusters and sparse clusters at the same time.

For the real data based on the text datasets, our task is to find dense clusters, which is consistent with the objectives of graph partitioning approaches. Overall, the SCC algorithms perform better than NC and METIS on the real datasets. Especially, SCC-ED provides the best performance in most datasets. The possible reasons for this are discussed as follows. First, the SCC model makes use of any possible block pattern in the relation matrices; on the other hand, the edge-cut based approaches focus on diagonal block patterns. Hence, the SCC model is more robust to heavily overlapping cluster structures. For example, for the difficult NG17-19 dataset, SCC algorithms do not totally fail as NC and METIS do. Second, since the edge weights from different graphs may have very different probabilistic distributions, popular Euclidean distance function, which corresponds to normal distribution assumption, are not always appropriate. By Theorem 1, edge-cut based algorithms are based on Euclidean distance. On the other hand, SCC-GI is based on generalized I-divergence corresponding to Poisson distribution assumption, which is more appropriate for graphs based on text data. Note that how to choose distance functions for specific graphs is non-trivial and beyond the scope of this study. Third, unlike METIS, the SCC algorithms do not restrict clusters to have an equal size and hence they are more robust to unbalanced clusters.

In the experiments, we observe that SCC algorithms performs stably and rarely provide unreasonable solution, though like other algorithms SCC algorithms provide local optima to the NP-hard clustering problem. In the experiments, we also observe that the order of the actual running time for the algorithms is consistent with

theoretical analysis, i.e., METIS<SCC<NC. For example, in a test run on NG1-20, METIS, SCC-ED, SCC-GI, and NC take 8.96, 11.4, 12.1, and 35.8 seconds, respectively. METIS is the best, since it is quasi-linear.

We also run the SCC-ED algorithm on the actor/actress graph based on IMDB movie dataset for a case study of social network analysis. We formulate a graph of 20000 nodes, in which each node represents an actors/actresses and the edges denote collaboration between them. The number of the cluster is set to be 200. Although there is no ground truth for the clusters, we observe that the results consist of a large number of interesting and meaningful clusters, such as clusters of actors with a similar style and tight clusters of the actors from a movie or a movie serial. For example, Table 7 shows Community 121 consisting of 21 actors/actresses, which contains the actors/actresses in movies series "The Lord of Rings".

Table 7 The members of cluster 121 in the actor graph

Cluster 121
Viggo Mortensen, Sean Bean, Miranda Otto, Ian Holm, Brad Dourif, Cate Blanchett, Ian McKellen ,Liv Tyler , David Wenham , Christopher Lee, John Rhys-Davies , Elijah Wood , Bernard Hill, Sean Astin, Dominic Monaghan, Andy Serkis, Karl Urban , Orlando Bloom , Billy Boyd ,John Noble, Sala Baker

4 Special Homogeneous Relational Data – Documents with Citations

One of the most popular scenarios for link-based clustering is document clustering. Here textual documents form a special case of the general homogeneous relational data scenario, in which a document links to another one through a citation. In this section, we showcase how to use a generative model, a specific topic model, to solve for the document clustering problem.

By capturing the essential characteristics in documents, one gives documents a new representation, which is often more parsimonious and less noise-sensitive. Among the existing methods that extract essential characteristics from documents, topic model plays a central role. Topic models extract a set of latent topics from a corpus and as a consequence represent documents in a new latent semantic space. One of the well-known topic models is the Probabilistic Latent Semantic Indexing (PLSI) model proposed by Hofmann [29]. In PLSI each document is modeled as a probabilistic mixture of a set of topics. Going beyond PLSI, Blei et al. [5] presented the Latent Dirichlet Allocation (LDA) model by incorporating a prior for the topic distributions of the documents. In these probabilistic topic models, one assumption underpinning the generative process is that the documents are independent. However, this assumption does not always hold true in practice, because doc-

uments in a corpus are usually related to each other in certain ways. Very often, one can explicitly observe such relations in a corpus, e.g., through the citations and co-authors of a paper. In such a case, these observations should be incorporated into topic models in order to derive more accurate latent topics that better reflect the relations among the documents.

In this section, we present a generative model [25], called the *citation-topic* (CT) model for modeling linked documents that explicitly considers the relations among documents. In this model, the content of each document is a mixture of two sources: (1) the topics of the given document and (2) the topics of the documents that are related to (e.g., cited by) the given document. This perspective actually reflects the process of writing a scientific article: the authors probably first learn knowledge from the literature and then combine their own creative ideas with the learned knowledge to form the content of the paper. Furthermore, to capture the indirect relations among documents, CT contains a generative process to select related documents where the related documents are not necessarily directly linked to the given document. CT is applied to the document clustering task and the experimental comparisons against several state-of-the-art approaches demonstrate very promising performances.

4.1 Model Formulation and Algorithm

Suppose that the corpus consists of N documents $\{d_j\}_{j=1}^{N}$ in which M distinct words $\{w_i\}_{i=1}^{M}$ occur. Each document d might have a set of citations C_d, and thus the documents are linked together by these citations.

CT assumes the following generative process for each word w in the document d in the corpus.

1. Choose a related document c from $p(c|d,\Xi)$, a multinomial probability conditioned on the document d.
2. Choose a topic z from the topic distribution of the document c, $p(z|c,\Theta)$.
3. Choose a word w which follows the multinomial distribution $p(w|z,\Psi)$ conditioned on the topic z.

As a result, one obtains the observed pair (d,w), while the latent random variables c,z are discarded. To obtain a document d, one repeats this process $|d|$ times, where $|d|$ is the length of the document d. The corpus is obtained once every document in the corpus is generated by this process, as shown in Figure 5. In this generative model, the dimensionality K of the topic variable z is assumed known and the document relations are parameterized by an $N \times N$ matrix Ξ where $\Xi_{lj} = p(c = l|d = j)$, which is computed from the citation information of the corpus.

Following the maximum likelihood principle, one estimates the parameters by maximizing the log-likelihood function

$$\mathscr{L} = \sum_{j=1}^{N}\sum_{i=1}^{M} n(w_i,d_j)\log p(w_i|d_j) \tag{7}$$

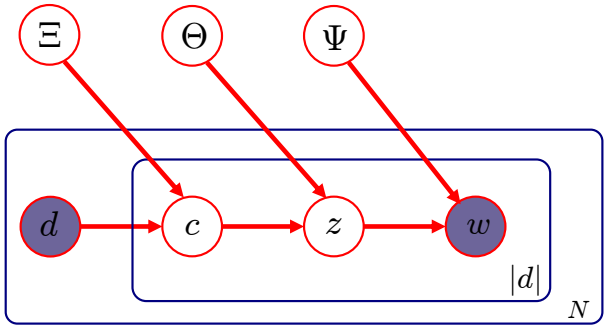


Fig. 5 CT using the plate notation.

where $n(w_i, d_j)$ denotes the number of the times w_i occurs in d_j. According to the above generative process, the log-likelihood function can be rewritten as the following equation

$$\mathscr{L} = \sum_{j=1}^{N}\sum_{i=1}^{M} n(w_i, d_j) \log\{\sum_{l=1}^{K}\sum_{h=1}^{N} p(w_i|z_l)p(z_l|d_h)p(d_h|d_j)\} \tag{8}$$

The expectation-maximization (EM) algorithm can be applied to estimate the parameters.

The document relation matrix Ξ is computed from the citation information of the corpus. Suppose that the document d_j has a set of citations Q_{d_j}. A matrix S is constructed to denote the direct relationships among the documents as follows: $S_{lj} = 1/|Q_{d_j}|$ for $d_l \in Q_{d_j}$ and 0 otherwise, where $|Q_{d_j}|$ denotes the size of the set Q_{d_j}. A simple method to obtain Ξ is to set $\Xi = \mathrm{S}$. However, this strategy only captures *direct* relations among the documents and overlooks *indirect* relationships. To better capture this transitive property, we choose a related document by a random walk on the directed graph represented by S. The probability that the random walk stops at the current node (and therefore chooses the current document as the related document) is specified by a parameter α. According to the properties of random walk, Ξ can be obtained by $\Xi = (1-\alpha)(\mathrm{I} - \alpha\mathrm{S})^{-1}$. The specific algorithm refers to [25].

4.2 Experiments

The experimental evaluations are reported on the document clustering task for a standard dataset Cora with the citation information available. Cora [41] contains the papers published in the conferences and journals of the different research areas in computer science, such as artificial intelligence, information retrieval, and hardware. A unique label has been assigned to each paper to indicate the research area it

belongs to. These labels serve as the ground truth in our performance studies. In the Cora dataset, there are 9998 documents where 3609 distinct words occur.

By representing documents in terms of latent topic space, topic models can assign each document to the most probable latent topic according to the topic distributions of the documents. For the evaluation purpose, CT is compared with the following representative clustering methods.

1. Traditional K-means.
2. Spectral Clustering with Normalized Cuts (Ncut) [43].
3. Nonnegative Matrix Factorization (NMF) [50].
4. Probabilistic Latent Semantic Indexing (PLSI) [29].
5. Latent Dirichlet Allocation (LDA) [5].
6. PHITS [12].
7. PLSI+PHITS, which corresponds to $\alpha = 0.5$ in [13].

The same evaluation strategy is used as in [50] for the clustering performance. The test data used for evaluating the clustering methods are constructed by mixing the documents from multiple clusters randomly selected from the corpus. The evaluations are conducted for different numbers of clusters K. At each run of the test, the documents from a selected number K of clusters are mixed, and the mixed document set, along with the cluster number K, is provided to the clustering methods. For each given cluster number K, 20 test runs are conducted on different randomly chosen clusters, and the final performance scores are obtained by averaging the scores over the 20 test runs.

The parameter α is simply fixed at 0.99 for the CT model. The accuracy comparisons with various numbers of clusters are reported in Figure 6, which shows that CT has the best performance in terms of the accuracy and the relationships among the documents do offer help in the document clustering.

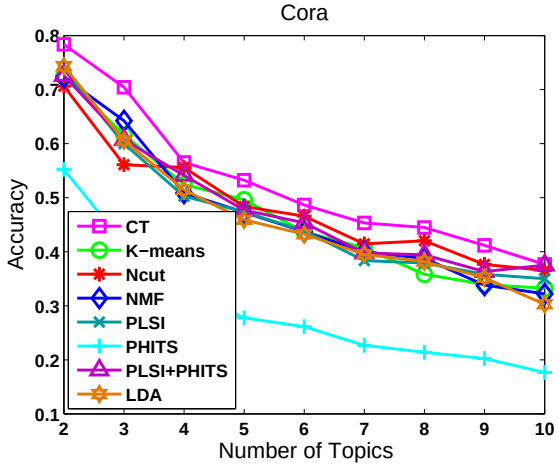


Fig. 6 Accuracy comparisons (the higher, the better).

5 General Relational Clustering through a Probabilistic Generative Model

In this section, as another example of a generative model in machine learning, we present a probabilistic generative framework to the general relational clustering. As mentioned before, in general, relational data contain three types of information, attributes for individual objects, homogeneous relations between objects of the same type, and heterogeneous relations between objects of different types. For example, for a scientific publication relational dataset of papers and authors, the personal information such as affiliation for authors is the attributes; the citation relations among papers are homogeneous relations; the authorship relations between papers and authors are heterogeneous relations. Such data violate the classic IID assumption in machine learning and statistics and present huge challenges to traditional clustering approaches. In Section 2, we have also shown that an intuitive solution to transform relational data into flat data and then to cluster each type of objects independently may not work. Moreover, a number of important clustering problems, which have been of intensive interest in the literature, can be viewed as special cases of the general relational clustering. For example, graph clustering (partitioning) [7, 9, 20, 27, 31, 43] can be viewed as clustering on singly-type relational data consisting of only homogeneous relations (represented as a graph affinity matrix); co-clustering [1, 15] which arises in important applications such as document clustering and micro-array data clustering, can be formulated as clustering on bi-type relational data consisting of only heterogeneous relations. Recently, semi-supervised clustering [3, 47] has attracted significant attention, which is a special type of clustering using both labeled and unlabeled data. In [38], it is shown that semi-supervised clustering can be formulated as clustering on singly-type relational data consisting of attributes and homogeneous relations.

Therefore, relational data present not only huge challenges to traditional unsupervised clustering approaches, but also great need for theoretical unification of various clustering tasks. In this section, we present a probabilistic framework for general relational clustering [38], which also provides a principal framework to unify various important clustering tasks including traditional attributes-based clustering, semi-supervised clustering, co-clustering, and graph clustering. The framework seeks to identify cluster structures for each type of data objects and interaction patterns between different types of objects. It is applicable to relational data of various structures. Under this framework, two parametric hard and soft relational clustering algorithms are developed under a large number of exponential family distributions. The algorithms are applicable to various relational data from various applications and at the same time unify a number of stat-of-the-art clustering algorithms: co-clustering algorithms, the k-partite graph clustering, Bregman k-means, and semi-supervised clustering based on hidden Markov random fields.

5.1 Model Formulation and Algorithms

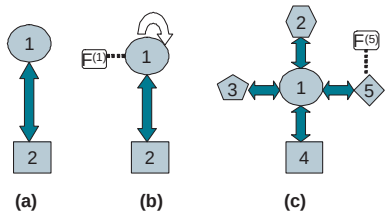


Fig. 7 Examples of the structures of relational data.

With different compositions of three types of information, attributes, homogeneous relations, and heterogeneous relations, relational data could have very different structures. Figure 7 shows three examples of the structures of relational data. Figure 7(a) refers to a simple bi-type of relational data with only heterogeneous relations such as word-document data. Figure 7(b) represents bi-type data with all types of information, such as actor-movie data, in which actors (type 1) have attributes such as gender; actors are related to each other by collaboration in movies (homogeneous relations); actors are related to movies (type 2) by taking roles in movies (heterogeneous relations). Figure 7(c) represents the data consisting of companies, customers, suppliers, shareholders and advertisement media, in which customers (type 5) have attributes.

In this study, a relational dataset is represented as a set of matrices. Assume that a relational dataset has m different types of data objects, $\mathscr{X}^{(1)} = \{x_i^{(1)}\}_{i=1}^{n_1}, \ldots, \mathscr{X}^{(m)} = \{x_i^{(m)}\}_{i=1}^{n_m}$, where n_j denotes the number of objects of the jth type and $x_p^{(j)}$ denotes the name of the pth object of the jth type. The observations of the relational data are represented as three sets of matrices, attribute matrices $\{\mathbf{F}^{(j)} \in \mathbb{R}^{d_j \times n_j}\}_{j=1}^{m}$, where d_j denotes the dimension of attributes for the jth type objects and $\mathbf{F}_{\cdot p}^{(j)}$ denotes the attribute vector for object $x_p^{(j)}$; homogeneous relation matrices $\{\mathbf{S}^{(j)} \in \mathbb{R}^{n_j \times n_j}\}_{j=1}^{m}$, where $\mathbf{S}_{pq}^{(j)}$ denotes the relation between $x_p^{(j)}$ and $x_q^{(j)}$; heterogeneous relation matrices $\{\mathbf{R}^{(ij)} \in \mathbb{R}^{n_i \times n_j}\}_{i,j=1}^{m}$, where $\mathbf{R}_{pq}^{(ij)}$ denotes the relation between $x_p^{(i)}$ and $x_q^{(j)}$. The above representation is a general formulation. In real applications, not every type of objects has attributes, homogeneous relations, and heterogeneous relations all together. For example, the relational dataset in Figure 7(a) is represented by only one heterogeneous matrix $\mathbf{R}^{(12)}$, and the one in Figure 7(b) is represented by three matrices, $\mathbf{F}^{(1)}$, $\mathbf{S}^{(1)}$ and $\mathbf{R}^{(12)}$. Moreover, for a specific clustering task, we may not use all available attributes and relations after feature or relation selection pre-processing.

Mixed membership models, which assume that each object has mixed membership denoting its association with classes, have been widely used in the applications involving soft classification [21], such as matching words and pictures [6], race genetic structures [6, 48], and classifying scientific publications [22]. Consequently, a

relational mixed membership model is developed to cluster relational data (which is referred to *mixed membership relational clustering* or MMRC throughout the rest of the section).

Assume that each type of objects $\mathscr{X}^{(j)}$ has k_j latent classes. We represent the membership vectors for all the objects in $\mathscr{X}^{(j)}$ as a membership matrix $\Lambda^{(j)} \in [0,1]^{k_j \times n_j}$ such that the sum of elements of each column $\blacksquare_{\cdot p}^{(j)}$ is 1 and $\blacksquare_{\cdot p}^{(j)}$ denotes the membership vector for object $x_p^{(j)}$, i.e., $\blacksquare_{gp}^{(j)}$ denotes the probability that object $x_p^{(j)}$ associates with the gth latent class. We also write the parameters of distributions to generate attributes, homogeneous relations, and heterogeneous relations in matrix forms. Let $\Theta^{(j)} \in \mathbb{R}^{d_j \times k_j}$ denote the distribution parameter matrix for generating attributes $\mathbf{F}^{(j)}$ such that $\Theta_{\cdot g}^{(j)}$ denotes the parameter vector associated with the gth latent class. Similarly, $\Gamma^{(j)} \in \mathbb{R}^{k_j \times k_j}$ denotes the parameter matrix for generating homogeneous relations $\mathbf{S}^{(j)}$; $\Upsilon^{(ij)} \in \mathbb{R}^{k_i \times k_j}$ denotes the parameter matrix for generating heterogeneous relations $\mathbf{R}^{(ij)}$. In summary, the parameters of MMRC model are

$$\Omega = \{\{\Lambda^{(j)}\}_{j=1}^m, \{\Theta^{(j)}\}_{j=1}^m, \{\Gamma^{(j)}\}_{j=1}^m, \{\Upsilon^{(ij)}\}_{i,j=1}^m\}.$$

In general, the meanings of the parameters, Θ, Λ, and Υ, depend on the specific distribution assumptions. However, in [38], it is shown that for a large number of exponential family distributions, these parameters can be formulated as expectations with intuitive interpretations.

Next, we introduce the latent variables into the model. For each object x_p^j, a latent cluster indicator vector is generated based on its membership parameter $\Lambda_{\cdot p}^{(j)}$, which is denoted as $\mathbf{C}_{\cdot p}^{(j)}$, i.e., $\mathbf{C}^{(j)} \in \{0,1\}^{k_j \times n_j}$ is a latent indicator matrix for all the jth type objects in $\mathscr{X}^{(j)}$.

Finally, we present the generative process of observations, $\{\mathbf{F}^{(j)}\}_{j=1}^m$, $\{\mathbf{S}^{(j)}\}_{j=1}^m$, and $\{\mathbf{R}^{(ij)}\}_{i,j=1}^m$ as follows:

1. For each object $x_p^{(j)}$
   - Sample $\mathbf{C}_{\cdot p}^{(j)} \sim Multinomial(\Lambda_{\cdot p}^{(j)}, 1)$.
2. For each object $x_p^{(j)}$
   - Sample $\mathbf{F}_{\cdot p}^{(j)} \sim Pr(\mathbf{F}_{\cdot p}^{(j)} | \Theta^{(j)} \mathbf{C}_{\cdot p}^{(j)})$.
3. For each pair of objects $x_p^{(j)}$ and $x_q^{(j)}$
   - Sample $\mathbf{S}_{pq}^{(j)} \sim Pr(\mathbf{S}_{pq}^{(j)} | (\mathbf{C}_{\cdot p}^{(j)})^T \Gamma^{(j)} \mathbf{C}_{\cdot q}^{(j)})$.
4. For each pair of objects $x_p^{(i)}$ and $x_q^{(j)}$
   - Sample $\mathbf{R}_{pq}^{(ij)} \sim Pr(\mathbf{R}_{pq}^{(ij)} | (\mathbf{C}_{\cdot p}^{(i)})^T \Upsilon^{(ij)} \mathbf{C}_{\cdot q}^{(j)})$.

In the above generative process, a latent indicator vector for each object is generated based on multinomial distribution with the membership vector as parameters. Ob-

servations are generated independently conditioning on latent indicator variables. The parameters of condition distributions are formulated as products of the parameter matrices and latent indicators, i.e., $Pr(\mathbf{F}_{\cdot p}^{(j)}|\mathbf{C}_{\cdot p}^{(j)},\Theta^{(j)}) = Pr(F_{\cdot p}^{(j)}|\Theta^{(j)}\mathbf{C}_{\cdot p}^{(j)})$, $Pr(\mathbf{S}_{pq}^{(j)}|\mathbf{C}_{\cdot p}^{(j)},\mathbf{C}_{\cdot q}^{(j)},\Gamma^{(j)}) = Pr(\mathbf{S}_{pq}^{(j)}|(\mathbf{C}_{\cdot p}^{(j)})^T\Gamma^{(j)}\mathbf{C}_{\cdot q}^{(j)})$, and $Pr(\mathbf{R}_{pq}^{(ij)}|\mathbf{C}_{\cdot p}^{(i)},\mathbf{C}_{\cdot q}^{(j)},\Upsilon^{(ij)}) = Pr(\mathbf{R}_{pq}^{(ij)}|(\mathbf{C}_{\cdot p}^{(i)})^T\Upsilon^{(ij)}\mathbf{C}_{\cdot q}^{(j)})$. Under this formulation, an observation is sampled from the distributions of its associated latent classes. For example, if $\mathbf{C}_{\cdot p}^{(i)}$ indicates that $x_p^{(i)}$ is with the gth latent class and $\mathbf{C}_{\cdot q}^{(j)}$ indicates that $x_q^{(j)}$ is with the hth latent class, then $(\mathbf{C}_{\cdot p}^{(i)})^T\Upsilon^{(ij)}\mathbf{C}_{\cdot q}^{(j)} = \Upsilon_{gh}^{(ij)}$. Hence, we have $Pr(\mathbf{R}_{pq}^{(ij)}|\Upsilon_{gh}^{(ij)})$ implying that the relation between $x_p^{(i)}$ and $x_q^{(j)}$ is sampled by using the parameter $\Upsilon_{gh}^{(ij)}$.

With matrix representation, the joint probability distribution over the observations and the latent variables can be formulated as follows,

$$\begin{aligned} Pr(\Psi|\Omega) = \prod_{j=1}^{m} Pr(\mathbf{C}^{(j)}|\Lambda^{(j)}) \prod_{j=1}^{m} Pr(\mathbf{F}^{(j)}|\Theta^{(j)}\mathbf{C}^{(j)}) \\ \prod_{j=1}^{m} Pr(\mathbf{S}^{(j)}|(\mathbf{C}^{(j)})^T\Gamma^{(j)}\mathbf{C}^{(j)}) \prod_{i=1}^{m}\prod_{j=1}^{m} Pr(\mathbf{R}^{(ij)}|(\mathbf{C}^{(i)})^T\Upsilon^{(ij)}\mathbf{C}^{(j)}) \end{aligned} \tag{9}$$

where $\Psi = \{\{\mathbf{C}^{(j)}\}_{j=1}^m, \{\mathbf{F}^{(j)}\}_{j=1}^m, \{\mathbf{S}^{(j)}\}_{j=1}^m, \{\mathbf{R}^{(ij)}\}_{i,j=1}^m\}$,
$Pr(\mathbf{C}^{(j)}|\Lambda^{(j)}) = \prod_{p=1}^{n_j} Multinomial(\Lambda_{\cdot p}^{(j)}, 1)$,
$Pr(\mathbf{F}^{(j)}|\Theta^{(j)}\mathbf{C}^{(j)}) = \prod_{p=1}^{n_j} Pr(\mathbf{F}_{\cdot p}^{(j)}|\Theta^{(j)}\mathbf{C}_{\cdot p}^{(j)})$,
$Pr(\mathbf{S}^{(j)}|(\mathbf{C}^{(j)})^T\Gamma^{(j)}\mathbf{C}^{(j)}) = \prod_{p,q=1}^{n_j} Pr(\mathbf{S}_{pq}^{(j)}|(\mathbf{C}_{\cdot p}^{(j)})^T\Gamma^{(j)}\mathbf{C}_{\cdot q}^{(j)})$,
and similarly for $\mathbf{R}^{(ij)}$.

Based on the MMRC model, we are able to derive the soft version MMRC, the hard version MMRC, as well as the mixed version MMRC (i.e., the combination of the soft version and the hard version MMRC) algorithms under all the exponential family functions [38]. In addition, we also show that many existing models and algorithms in the literature are the variations or special cases of the MMRC model. Specifically, we have demonstrated this unified view to the classic attribute based clustering (including the k-means), the mixture model EM clustering, semi-supervised clustering, co-clustering, and graph clustering in the literature.

5.2 Experiments

This section provides empirical evidence to show the effectiveness of the MMRC model and algorithms. Since a number of stat-of-the-art clustering algorithms [1–3, 11, 15, 35] can be viewed as special cases of the MMRC model and algorithms, the experimental results in these efforts also illustrate the effectiveness of the MMRC model and algorithms. Here we apply MMRC algorithms to the tasks of

graph clustering, bi-clustering, tri-clusering, and clustering on a general relational dataset of all three types of information. In the experiments, we use mixed version MMRC, i.e., hard MMRC initialization followed by soft MMRC. Although MMRC can adopt various distribution assumptions, due to space limit, we use MMRC under normal or Poisson distribution assumption in the experiments. However, this does not imply that they are optimal distribution assumptions for the data. How to decide the optimal distribution assumption is beyond the scope of this study.

For performance measure, we elect to use the Normalized Mutual Information (NMI) [44] between the resulting cluster labels and the true cluster labels, which is a standard way to measure the cluster quality. The final performance score is the average of ten runs.

5.2.1 Graph Clustering

In this section, we present experiments on the MMRC algorithm under normal distribution in comparison with two representative graph partitioning algorithms, the spectral graph partitioning (SGP) from [42] that is generalized to work with both normalized cut and ratio association, and the classic multilevel algorithm, METIS [31].

The graphs based on the text data have been widely used to test graph partitioning algorithms [18, 20, 52]. In this study, we use various datasets from the 20-newsgroups [34], WebACE, and TREC [30], which cover datasets of different sizes, different balances, and different levels of difficulties. The data are pre-processed by removing the stop words and each document is represented by a term-frequency vector using TF-IDF weights. Then we construct relational data for each text dataset such that objects (documents) are related to each other with the cosine similarities between the term-frequency vectors. A summary of all the datasets to construct relational data used in this study is shown in Table 8, in which n denotes the number of objects in the relational data, k denotes the number of true clusters, and *balance* denotes the size ratio of the smallest clusters to the largest clusters.

Table 8 Summary of relational data for Graph Clustering.

Name	n	k	Balance	Source
tr11	414	9	0.046	TREC
tr23	204	6	0.066	TREC
NG1-20	14000	20	1.0	20-newsgroups
k1b	2340	6	0.043	WebACE

For the number of clusters k, we simply use the number of the true clusters. Note that how to choose the optimal number of clusters is a nontrivial model selection problem and beyond the scope of this study.

Figure 8 shows the NMI comparison of the three algorithms. We observe that although there is no single winner on all the graphs, overall the MMRC algorithm performs better than SGP and METIS. Especially on the difficult dataset tr23, MMRC

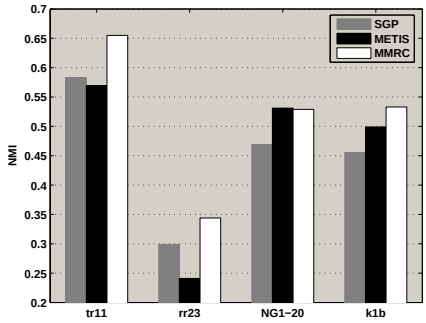


Fig. 8 NMI comparison of SGP, METIS, and MMRC algorithms.

increases the performance about 30%. Hence, MMRC under normal distribution provides a new graph partitioning algorithm which is viable and competitive compared with the two existing state-of-the-art graph partitioning algorithms. Note that although the normal distribution is most popular, MMRC under other distribution assumptions may be more desirable in specific graph clustering applications depending on the statistical properties of the graphs.

5.2.2 Bi-clustering and Tri-clustering

Table 9 Subsets of the 20-Newsgroups data for the bi-type relational data

Dataset Name	Newsgroups Included	# Documents per Group	Total # Documents
BT-NG1	rec.sport.baseball, rec.sport.hockey	200	400
BT-NG2	comp.os.ms-windows.misc, comp.windows.x, rec.motorcycles, sci.crypt, sci.space	200	1000
BT-NG3	comp.os.ms-windows.misc, comp.windows.x, misc.forsale, rec.motorcycles,rec.motorcycles,sci.crypt, sci.space, talk.politics.mideast, talk.religion.misc	200	1600

In this section, we apply the MMRC algorithm under Poisson distribution to clustering bi-type relational data, word-document data, and tri-type relational data, word-document-category data. Two algorithms, Bi-partite Spectral Graph partitioning (BSGP) [18] and Relation Summary Network under Generalized I-divergence (RSN-GI) [39], are used as comparison in bi-clustering. For tri-clustering, Consistent Bipartite Graph Co-partitioning (CBGC) [24] and RSN-GI are used as comparison.

The bi-type relational data, word-document data, are constructed based on various subsets of the 20-Newsgroup data. We pre-process the data by selecting the top 2000 words by the mutual information. The document-word matrix is based on *tf.idf*

Table 10 Taxonomy structures of the two datasets for constructing tri-partite relational data

Dataset	Taxonomy structure
TT-TM1	{rec.sport.baseball, rec.sport.hockey}, {talk.politics.guns, talk.politics.mideast, talk.politics.misc}
TT-TM2	{comp.graphics, comp.os.ms-windows.misc}, {rec.autos, rec.motorcycles}, {sci.crypt, sci.electronics}

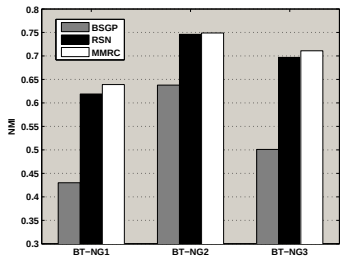


Fig. 9 NMI comparison of BSGP, RSN and MMRC algorithms for bi-type data.

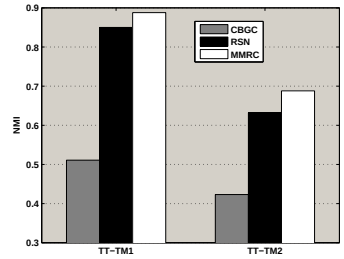


Fig. 10 NMI comparison of CBGC, RSN and MMRC algorithms for tri-type data.

weighting scheme and each document vector is normalized to a unit L_2 norm vector. Specific details of the datasets are listed in Table 9. For example, for the dataset *BT-NG3* we randomly and evenly sample 200 documents from the corresponding newsgroups; then we formulate a bi-type relational data set of 1600 document and 2000 word.

The tri-type relational data are built based on the 20-newsgroups data for hierarchical taxonomy mining. In the field of text categorization, hierarchical taxonomy classification is widely used to obtain a better trade-off between effectiveness and efficiency than flat taxonomy classification. To take advantage of hierarchical classification, one must mine a hierarchical taxonomy from the dataset. We see that words, documents, and categories formulate a sandwich structure tri-type relational dataset, in which documents are the central type nodes. The links between documents and categories are constructed such that if a document belongs to k categories, the weights of links between this document and these k category nodes are $1/k$ (re-

fer to [24] for details). The true taxonomy structures for the two datasets, *TP-TM1* and *TP-TM2*, are documented in Table 10.

Figure 9 and Figure 10 show the NMI comparison of the three algorithms on bi-type and tri-type relational data, respectively. We observe that the MMRC algorithm performs significantly better than BSGP and CBGC. MMRC performs slightly better than RSN on some datasets. Since RSN is a special case of hard MMRC, this shows that mixed MMRC improves hard MMRC's performance on the datasets. Therefore, compared with the existing state-of-the-art algorithms, the MMRC algorithm performs more effectively on these bi-clustering or tri-clustering tasks and on the other hand, it is flexible for different types of multi-clustering tasks which may be more complicated than tri-type clustering.

5.2.3 A Case Study on Actor-movie Data

Table 11 Two clusters from actor-movie data

cluster 23 of actors
Viggo Mortensen, Sean Bean, Miranda Otto, Ian Holm, Christopher Lee, Cate Blanchett, Ian McKellen ,Liv Tyler , David Wenham , Brad Dourif , John Rhys-Davies , Elijah Wood , Bernard Hill , Sean Astin , Andy Serkis , Dominic Monaghan , Karl Urban , Orlando Bloom , Billy Boyd ,John Noble, Sala Baker
cluster 118 of movies
The Lord of the Rings: The Fellowship of the Ring (2001) The Lord of the Rings: The Two Towers (2002) The Lord of the Rings: The Return of the King (2003)

We also run the MMRC algorithm on the actor-movie relational data based on the IMDB movie dataset for a case study. In the data, actors are related to each other by collaboration (homogeneous relations); actors are related to movies by taking roles in the movies (heterogeneous relations); movies have attributes such as release time and rating (note that there are no links between movies). Hence the data have all the three types of information. We formulate a dataset of 20000 actors and 4000 movies. We run experiments with $k = 200$. Although there is no ground truth for the data's cluster structure, we observe that most resulting clusters that are actors or movies of the similar style such as action, or tight groups from specific movie serials. For example, Table 11 shows cluster 23 of actors and cluster 118 of movies; the parameter $\Upsilon_{23,118}$ shows that these two clusters are strongly related to each other. In fact, the actor cluster contains the actors in the movie series "The Lord of the Rings". Note that if we only have one type of actor objects, we only get the actor clusters, but with two types of nodes, although there is no link between the movies, we also get the related movie clusters to explain how the actors are related.

6 Dynamic Relational Data Clustering through Graphical Models

We have studied extensively on static relational data clustering in the previous sections. In this section, we switch our focus to dynamic scenarios. One popular example of the dynamic scenarios is the evolutionary clustering. Evolutionary clustering is a recently identified new and hot research topic in data mining. Evolutionary clustering addresses the evolutionary trend development regarding a collection of data items that evolves over the time. From time to time, with the evolution of the data collection, new data items may join the collection and existing data items may leave the collection; similarly, from time to time, cluster structure and cluster number may change during the evolution. Due to the nature of the evolution, model selection must be solved as part of a solution to the evolutionary clustering problem at each time. Consequently, evolutionary clustering poses a greater challenge than the classic, static clustering problem as many existing solutions to the latter problem typically assume that the model selection is still an open problem in the clustering literature.

In evolutionary clustering, one of the most difficult and challenging issues is to solve the correspondence problem. The correspondence problem refers to the correspondence between different local clusters across the times due to the evolution of the distribution of the clusters, resulting in cluster-cluster correspondence and cluster transition correspondence issues. All the existing methods in the literature fail to address the correspondence problems explicitly.

On the other hand, solutions to the evolutionary clustering problem have found a wide spectrum of applications for trend development analysis, social network evolution analysis, and dynamic community development analysis. Potential and existing applications include daily news analysis to observe news focus change, blog analysis to observe community development, and scientific publications analysis to identify the new and hot research directions in a specific area. Consequently, evolutionary clustering has recently become a very hot and focused research topic.

In this study [49], we show a new statistical graphical model HDP-HTM that we have developed as an effective solution to the evolutionary clustering problem. In this new model, we assume that the cluster structure at each time is a mixture model of the clusters for the data collection at that time; in addition, clusters at different times may share common clusters, resulting in explicitly addressing the cluster-cluster correspondence issue. we adopt the Hierarchical Dirichlet Processes (HDP) [46] with a set of common clusters at the top level of the hierarchy and the local clusters at the lower level at different times sharing the top level clusters. Further, data and clusters evolve over the time with new clusters and new data items possibly joining the collection and with existing clusters and data items possibly leaving the collection at different times, leading to the cluster structure and the number of clusters evolving over the time. Here, we use the state transition matrix to explicitly reflect the cluster-to-cluster transitions between different times, resulting in explicitly effective solution to the cluster transition correspondence issue. Consequently,

we propose the Infinite Hierarchical Hidden Markov State model (iH^{2}MS) to construct the Hierarchical Transition Matrix (HTM) at different times to capture the cluster-to-cluster transition evolution.

7 Infinite Hierarchical Hidden Markov State Model (iH^{2}MS)

Here, we present a new infinite hierarchical hidden Markov state model (iH^{2}MS) for Hierarchical Transition Matrix (HTM) and provide an update construction scheme based on this model. Figure 11 illustrates this model.

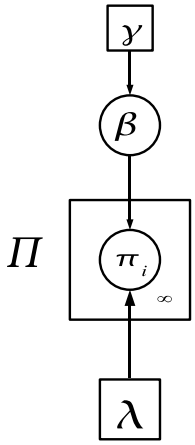


Fig. 11 The iH^{2}MS Model

Traditionally, Hidden Markov Model (HMM) has a *finite* state space with K hidden states, say $\{1,2,\ldots K\}$. For the hidden state sequence $\{s_1,s_2,\ldots s_T\}$ up to time T, there is a K by K state transition probability matrix Π governed by Markov dynamics with all the elements $\pi_{i,j}$ of each row π_i summed to 1.

$$\pi_{i,j} = p(s_t = j | s_{t-1} = i)$$

The initial state probability for state i is $p(s_1 = i)$ with the summation of all the initial probabilities equal to 1. For observation x_t in the observation sequence $\{x_1,x_2,\ldots x_T\}$, given state $s_t \in \{1,2,\ldots,K\}$, there is a parameter ϕ_{s_t} drawn from the base measure H which parameterizes the observation likelihood probability.

$$x_t | s_t \sim F(\phi_{s_t})$$

However, when dealing with a countable *infinite* state space, $\{1,2,\ldots K,\ldots\}$, we must adopt a new model similar to that in [4] for a state transition probability matrix with an *infinite* matrix dimension. Thus, the dimension of the state transition probability matrix now has become infinite. π_i, the i-th row of the transition probability matrix Π, may be represented as the mixing proportions for all the next infinite states, given the current state. Thus, we model it as a Dirichlet process (DP) with an infinite dimension with the summation of all the elements in a row normalized to 1, which leads to an infinite number of DPs' construction for an infinite transition probability matrix.

With no further prior knowledge on the state sequence, a typical prior for the transition probability may be the symmetric Dirichlet distributions. Similar to [46], we intend to construct a hierarchical Dirichlet model to keep different rows of the transition probability matrix to share part of the prior mixing proportions of each state at the top level. Consequently, we adopt a new state model, Infinite Hierarchical Hidden Markov State model (iH^{2}MS), to construct the Infinite Transition Probability Matrix which is called the Hierarchical Transition Matrix (HTM).

Similar to HDP [46], we draw a random probability measure on the infinite state space β as the top level prior from $stick(\gamma)$ represented as the mixing proportions of each state.

$$\beta = (\beta_k)_{k=1}^{\infty} \quad \beta_k = \beta_k{}' \prod_{l=1}^{k-1}(1-\beta_l{}') \quad \beta_k{}' \sim Beta(1,\gamma) \tag{10}$$

Here, the mixing proportion of state k, β_k, may also be interpreted as the prior mean of the transition probabilities leading to state k. Hence, β may be represented as the prior random measure of a transition probability DP.

For the ith row of the transition matrix Π, π_i, we sample it from $DP(\lambda,\beta)$ with a smaller concentration parameter λ implying a larger variability around the mean measure β. The stick-breaking representation for π_i is as follows:

$$\pi_i = (\pi_{i,k})_{k=1}^{\infty} \quad \pi_{i,k} = \pi_{i,k}{}' \prod_{l=1}^{k-1}(1-\pi_{i,l}{}') \quad \pi_{i,k}{}' \sim Beta(1,\lambda) \tag{11}$$

Specifically, $\pi_{i,k}$ is the state transition probability from the previous state i to the current state k as $p(s_t = k|s_{t-1} = i)$.

Now, each row of the transition probability matrix is represented as a DP which shares the same reasonable prior on the mixing proportions of the states. For a new row corresponding to a new state k, we simply draw a transition probability vector π_k from $DP(\lambda,\beta)$, resulting in constructing a countably infinite transition probability matrix. The transition probability constructed by iH^{2}MS may be further extended to the scenario where there are more than one state at each time [49]. HTM is estimated through the maximum likelihood principle [49].

7.1 Model Formulation and Algorithm

To capture the state (cluster) transition correspondence during the evolution at different times, we have proposed the HTM; at the same time, we must capture the state-state (cluster-cluster) correspondence, which may be handled by a hierarchical model with the top level corresponding to the global states[1] and the lower level corresponding to the local states, where it is natural to model the statistical process as HDP [46]. Consequently, we intend to combine HDP with HTM as a new HDP-HTM model, as illustrated in Figure 12.

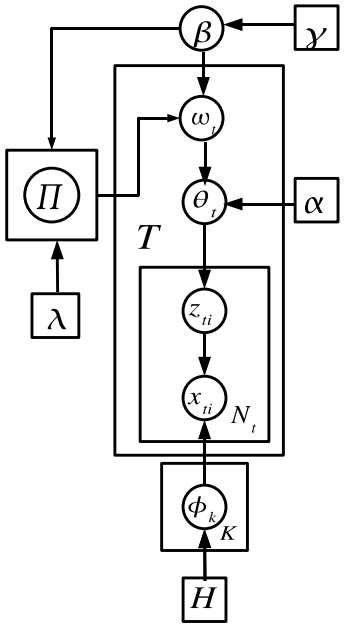


Fig. 12 The HDP-HTM Model

Let the global state space S denote the global cluster set, which includes all the states $S_t \subseteq S$ at all the times t. The global observation set X includes all the observations X_t at each time t, of which each data item i is denoted as $x_{t,i}$.

We draw the global mixing proportion from the global states β with the stick-breaking representation using the concentration parameter γ from (10). The global measure G_0 may be represented as:

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}$$

[1] Each state is represented as a distinct cluster.

where ϕ_k is drawn from the base probability measure H with pdf h, and δ_{ϕ_k} is the concentration measure on ϕ_k.

Different from HDP, here we must consider the evolution of the data and the states (i.e., the clusters). The distribution of the clusters at time t is not only governed by the global measure G_0, but also is controlled by the data and cluster evolution in the history. Consequently, we make an assumption that the data and the clusters at time t are generated from the previous data and clusters, according to the mixture proportions of each cluster and the transition probability matrix. The global prior mixture proportions for the clusters are β, and the state transition matrix Π provides the information of the previous state evolution in the history up to time t. Now, the expected number of the data items generated by cluster k is proportional to the number of data items in the clusters in the history multiplied by the transition probabilities from these clusters to state k; specificially, the mean mixture proportion for cluster k at time t, ω_t, is defined as follows:

$$\omega_{t,k} = \sum_{j=1}^{\infty} \beta_j \pi_{j,k}$$

More precisely, ω_t is further obtained by:

$$\omega_t = \beta \cdot \Pi \tag{12}$$

Clearly, by the transition probability property, $\sum_{k=1}^{\infty} \omega_{t,k} = 1$, $\sum_{k=1}^{\infty} \pi_{i,k} = 1$ and the stick-breaking property $\sum_{j=1}^{\infty} \beta_j = 1$.

$$\sum_{k=1}^{\infty} \omega_{t,k} = \sum_{k=1}^{\infty} \sum_{j=1}^{\infty} \beta_j \pi_{j,k} = \sum_{j=1}^{\infty} \beta_j \sum_{k=1}^{\infty} \pi_{j,k} = \sum_{j=1}^{\infty} \beta_j = 1$$

Thus, the mean mixture proportion ω_t may be taken as the new probability measure at time t on the global cluster set. With the concentration parameter α, we draw the mixture proportion vector θ_t from $DP(\alpha, \omega_t)$.

$$\theta_t | \alpha, \omega_t \sim DP(\alpha, \omega_t)$$

Now, at time t, the local measure G_t shares the global clusters parameterized by $\phi = (\phi_k)_{k=1}^{\infty}$ with the mixing proportion vector θ_t.

$$G_t = \sum_{k=1}^{\infty} \theta_{t,k} \delta_{\phi_k}$$

At time t, given the mixture proportion of the clusters θ_t, we draw a cluster indicator $z_{t,i}$ for data item $x_{t,i}$ from a multinomial distribution:

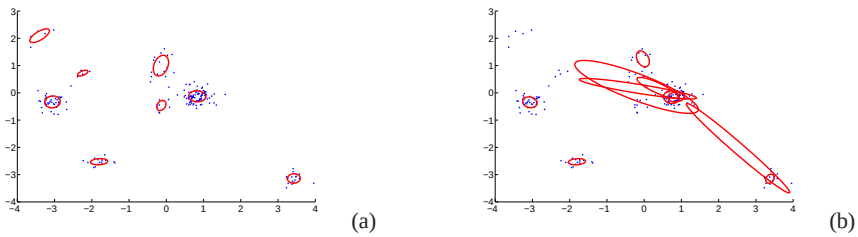
$$z_{t,i} | \theta_t \sim Mult(\theta_t)$$

Once we have the cluster indicator $z_{t,i}$, data item $x_{t,i}$ may be drawn from distribution F with pdf f, parameterized by ϕ from the base measure H.

$$x_{t,i}|z_{t,i},\phi \sim f(x|\phi_{z_{t,i}})$$

Finally, we summarize the data generation process for HDP-HTM as follows.

1. Sample the cluster parameter vector ϕ from the base measure H. The number of the parameters is unknown *a priori*, but is determined by the data when a new cluster is needed.
2. Sample the global cluster mixture vector β from $stick(\gamma)$.
3. At time t, compute the mean measure ω_t for the global cluster set by β and Π according to (12).
4. At time t, sample the local mixture proportion θ_t by $DP(\alpha,\omega_t)$.
5. At time t, sample the cluster indicator $z_{t,i}$ from $Mult(\theta_t)$ for data item $x_{t,i}$.
6. At time t, sample data item $x_{t,i}$ from $f(x|\phi_{z_{t,i}})$ given cluster indicator $z_{t,i}$ and parameter vector ϕ.



(a) (b)

Fig. 13 Illustrated Clustering results of HDP-HTM (a) and PCQ (b) for the synthetic data

7.2 Experiments

We have evaluated the HDP-HTM model in an extensive scale against the state-of-the-art literature. We compare HDP-HTM in performance with evolutionary spectral clustering PCM and PCQ algorithms [10] and HDP [46] for the synthetic data and the real data in the application of document evolutionary clustering; for the experiments in text data evolutionary clustering, we have also evaluated the HDP-HTM model in comparison with LDA [5, 26] in addition. In particular, the evaluations are performed in three datasets, a synthetic dataset, the 20 Newsgroups dataset, and a Google daily news dataset we have collected over a period of 5 continuous days.

7.2.1 Synthetic Dataset

We have generated a synthetic dataset in a scenario of evolutionary development. The data are a collection of mixture models with the number of the clusters unknown *a priori* with a smooth transition over the time during the evolution. Specifically, we simulate the scenario of the evolution over 10 different times with each time's collection according to a DP mixture model with 200 2-dimensional Gaussian distribution points. 10 Gaussian points in $\mathbf{N}(\mathbf{0}, \mathbf{2I})$ are set as the 10 global clusters' mean parameters. Then 200 Gaussian points within a cluster are sampled with this cluster's mean parameter and deviation parameter sampling from $\mathbf{N}(\mathbf{0}, \mathbf{0.2I})$, where $\mathbf{I}$ is the identify matrix. After the generation of such a dataset, we obtain the number of the clusters and the cluster assignments as the ground truth. We intentionally generate different numbers of the clusters at different times, as shown in Figure 15.

In the inference process, we tune the hyperparameters as follows. In each iteration, we use the vague Gamma priors [23] to update α, λ, and γ from $\Gamma(1,1)$. Figure 13 shows an example of the clustering results between HDP-HTM and PCQ at time 8 for the synthetic data. Clearly, HDP-HTM has a much better performance than PCQ in this synthetic data.

For a more systematic evaluation on this synthetic dataset, we use NMI (Normalized Mutual Information) [45] to quantitatively compare the clustering performances among all the four algorithms (HDP-HTM, HDP, PCM, and PCQ). NMI measures how much information two random distribution variables (computed clustering assignment and groundtruth clustering assignment) share, the larger the better with 1 as normalized maximum value. Figure 14 documents the performance comparison. From this figure, the average NMI values across the 10 times for HDP-HTM and HDP are 0.86 and 0.78, respectively, while those for PCQ and PCM are 0.70 and 0.71, respectively. HDP works worse than HDP-HTM for the synthetic data. The reason is that HDP model is unable to capture the cluster transition correspondence during the evolution among the data collections across the time in this case while HDP-HTM is able to explicitly solve for this correspondence problem; on the other hand, HDP still performs better than PCQ and PCM as HDP is able to learn the cluster number automatically during the evolution.

Since one of the advantages of the HDP-HTM model is to be able to learn the number of the clusters and the clustering structures during the evolution, we report this performance for HDP-HTM compared with HDP on this synthetic dataset in Figure 15. Here, we define the expected number of the clusters at each time as the average number of the clusters in all the posterior sampling iterations after the burn-in period. Thus, these numbers are not necessarily integers. Clearly, both models are able to learn the cluster numbers, with HDP-HTM having a better performance than HDP. Since both PCQ and PCM do not have this capability, they are not included in this evaluation.

7.2.2 Real Dataset

In order to showcase the performance of HDP-HTM model on real data applications, we apply HDP-HTM to a subset of the 20 Newsgroups data[2]. We intentionally set the number of the clusters at each time as the same number to accommodate the comparing algorithms PCQ and PCM which have this assumption of the same cluster number over the evolution. Also we select 10 clusters (i.e., topics) from the dataset (alt.atheism, comp.graphics, rec.autos, rec.sport.baseball, sci.crypt, sci.electronics, sci.med, sci.space, soc.religion.christian, talk.politics.mideast), with each having 100 documents. To "simulate" the corresponding 5 different times, we then split the dataset into 5 different collections, each of which has 20 documents randomly selected from the clusters. Thus, each collection at a time has 10 topics to generate words. We have preprocessed all the documents with the standard text processing for removing the stop words and stemming the remaining words.

To apply the HDP-HTM and HDP models, a symmetric Dirichlet distribution is used with the parameter 0.5 for the prior base distribution H. In each iteration, we update α, γ, and λ in HDP-HTM, from the gamma priors $\Gamma(0.1, 0.1)$. For LDA, α is set 0.1 and the prior distribution of the topics on the words is a symmetric Dirichlet distribution with concentration parameter 1. Since LDA only works for one data collection and requires a known cluster number in advance, we explicitly apply LDA to the data collection with the ground truth cluster number as input at each time.

Figure 16 reports the overall performance comparison among all the five methods using NMI metric again. Clearly HDP-HTM outperforms PCQ, PCM, HDP, and LDA at all the times; in particular, the difference is substantial for PCQ and PCM. Figure 17 further reports the performance on learning the cluster numbers at different times for HDP-HTM compared with HDP. Both models have a reasonble performance in automatically learning the cluster number at each time in comparison with the ground truth, with HDP-HTM having a clearly better performance than HDP in average.

In order to truly demonstrate the performance of HDP-HTM in comparison with the state-of-the-art literature on a real evolutionary clustering scenario, we have manually collected Google News articles for a continuous period of five days with both the data items (i.e., words in the articles) and the clusters (i.e., the news topics) evolving over the time. The evolutionary ground truth for this dataset is as follows. For each of the continuous five days, we have the number of the words, the number of the clusters, the number of the documents as (6113, 5, 50), (6356, 6, 60), (7063, 5, 50), (7762, 6, 60), and (8035, 6, 60), respectively. In order to accommodate the assumption of PCM and PCQ that the cluster number stays the same during the evolution, but at the same time in order to demonstrate the capability of HDP-HTM to automatically learn the cluster number at each evolutionary time, we intentionally set the news topic number (i.e., the cluster number) at each day's collection to have a small variation deviation during the evolution. Again, in order to compare the text

[2] http:kdd.ics.uci.edu/databases/20newsgroups/

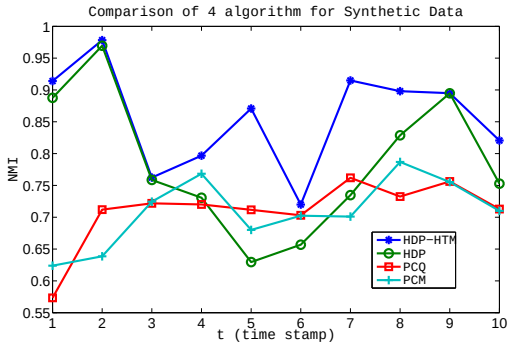


Fig. 14 The NMI performance comparison of the four algorithms on the synthetic dataset

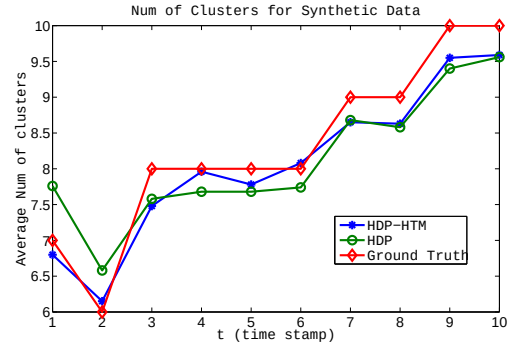


Fig. 15 The cluster number learning performance of the HDP-HTM in comparison with HDP on the synthetic dataset

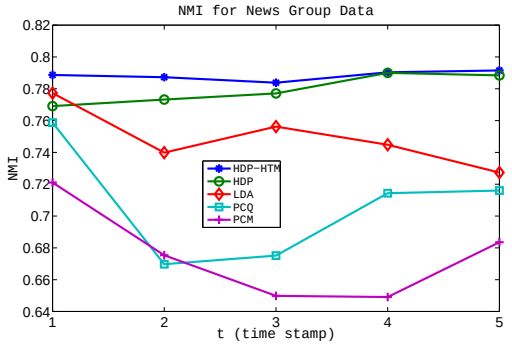


Fig. 16 The NMI performance comparison among the five algorithms on the 20 Newsgroups dataset

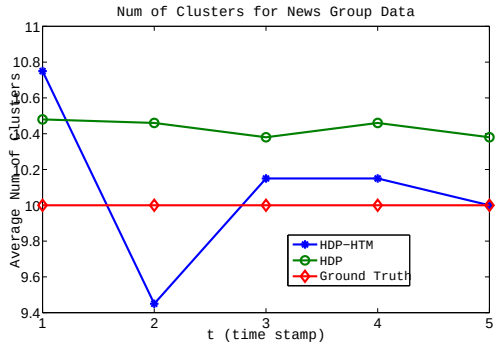


Fig. 17 Cluster number learning performance of HDP-HTM in comparison with HDP on the 20 Newsgroups dataset

clustering capability of LDA [5, 26] with a known topic number in advance, we use the ground truth cluster number at each time as the input to LDA. The parameter tuning process is similar to that in the experiment using the 20 Newsgroups dataset.

Figure 18 reports the NMI based performance evaluations among the five algorithms. Again, HDP-HTM outperforms PCQ, PCM, HDP, and LDA at all the times, especially substantially better than PCQ, PCM, and LDA. PCQ and PCM fail completely in most of the cases as they assume that the number of the clusters remains the same during the evolution, which is not true in this scenario.

Figure 19 further reports the performance on learning the cluster numbers for different times for HDP-HTM compared with HDP. In this dataset, HDP-HTM has a much better performance than HDP to learn the cluster numbers automatically at all the times.

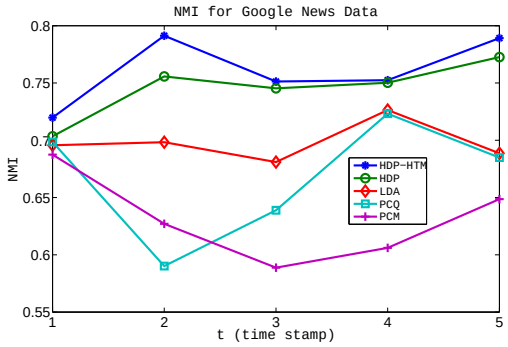


Fig. 18 The NMI performance comparison for all the five algorithms on the Google News dataset

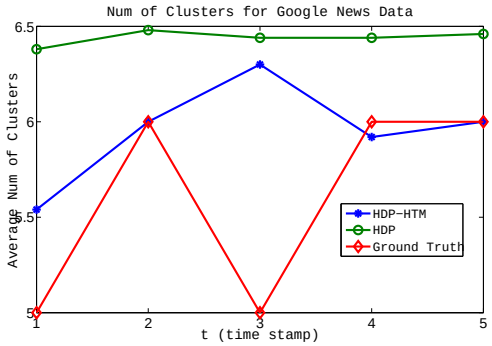


Fig. 19 The cluster number learning performance of HDP-HTM in comparison with HDP on the Google News dataset

8 Conclusions

In this chapter, we have reviewed several specific machine learning techniques used for different categories of link based or relational data clustering. Specifically, we have showcased a spectral clustering technique for heterogeneous relational clustering, a symmetric convex coding technique for homogeneous relational clustering, a citation model for the special homogeneous relational clustering — clustering textual documents with citations, a probabilistic generative model for general relational clustering, as well as a statistical graphical model for dynamic relational clustering. All these machine learning approaches are based on the mathematical foundation of matrix computation theory, probability, and statistics.

References

1. Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *KDD*, pages 509–514, 2004.
2. Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, 2005.
3. Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. ACM KDD04*, pages 59–68, Seattle, WA, August 2004.
4. Matthew J. Beal, Zoubin Ghahramani, and Carl Edward Rasmussen. The infinite hidden markov model. In *NIPS 14*, 2002.
5. D. M. Blei, A. Y. Ng, , and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.
6. D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.
7. Thang Nguyen Bui and Curt Jones. A heuristic for reducing fill-in in sparse matrix factorization. In *PPSC*, pages 445–452, 1993.

8. M. Catral, Lixing Han, Michael Neumann, and R.J. Plemmons. On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices. *Linear Algebra and Its Application*, 2004.
9. Pak K. Chan, Martine D. F. Schlag, and Jason Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *DAC '93*, pages 749–754, 1993.
10. Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162, 2007.
11. H. Cho, I. Dhillon, Y. Guan, and S. Sra. Minimum sum squared residue co-clustering of gene expression data. In *SDM*, 2004.
12. D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *Proc. ICML*, pages 167–174, 2000.
13. D. A. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Proc. NIPS*, pages 430–436, 2000.
14. D.D.Lee and H.S.Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
15. I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD'03*, pages 89–98, 2003.
16. Inderjit Dhillon, Yuqiang Guan, and Brian Kulis. A unified view of kernel k-means, spectral clustering and graph cuts. Technical Report TR-04-25, University of Texas at Austin, 2004.
17. Inderjit Dhillon, Yuqiang Guan, and Brian Kulis. A fast kernel-based multilevel algorithm for graph clustering. In *KDD '05*, 2005.
18. Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274, 2001.
19. Chris Ding, Xiaofei He, and H.D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM'05*, 2005.
20. Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of ICDM 2001*, pages 107–114, 2001.
21. E. Erosheva and S. E. Fienberg. Bayesian mixed membership models for soft clustering and classification. *Classification-The Ubiquitous Challenge*, pages 11–26, 2005.
22. E.A. Erosheva, S.E. Fienberg, and J. Lafferty. Mixed membership models of scientific publications. In *NAS*.
23. Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *The Annals of Statistics*, 90:577–588, 1995.
24. Bin Gao, Tie-Yan Liu, Xin Zheng, Qian-Sheng Cheng, and Wei-Ying Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *KDD '05*, pages 41–50, 2005.
25. Z. Guo, S. Zhu, Y. Chi, Z. Zhang, and Y. Gong. A latent topic model for linked documents. In *Proc. ACM SIGIR*, 2009.
26. Gregor Heinrich. Parameter estimation for text analysis. *Technical Report*, 2004.
27. Bruce Hendrickson and Robert Leland. A multilevel algorithm for partitioning graphs. In *Supercomputing '95*, page 28, 1995.
28. M. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. In *Proc. of the 18th International Joint Conference on Artificial Intelligence*, pages 1573–1579, 2003.
29. T. Hofmann. Probabilistic latent semantic indexing. In *Proc. SIGIR*, pages 50–57, 1999.
30. G. Karypis. A clustering toolkit, 2002.
31. George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.
32. B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970.
33. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks*, 31(11–16), 1999.

34. Ken Lang. News weeder: Learning to filter netnews. In *ICML*, 1995.
35. Tao Li. A general model for clustering binary data. In *KDD'05*, 2005.
36. B. Long, Zhang. Z., and P.S. Yu. Relational clustering by symmetric convex coding. In *Proc. International Conference on Machine Learning*, 2007.
37. B. Long, Z. Zhang, X. Wu, and P.S. Yu. Spectral clustering for multi-type relational data. In *Proc. ICML*, 2006.
38. B. Long, Z. Zhang, and P.S. Yu. A probabilistic framework for relational clustering. In *Proc. ACM KDD*, 2007.
39. Bo Long, Xiaoyun Wu, Zhongfei (Mark) Zhang, and Philip S. Yu. Unsupervised learning on k-partite graphs. In *KDD-2006*, 2006.
40. Bo Long, Zhongfei Mark Zhang, and Philip S. Yu. Co-clustering by block value decomposition. In *KDD'05*, 2005.
41. A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.
42. A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2001.
43. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.
44. Alexander Strehl and Joydeep Ghosh. Cluster ensembles – a knowledge reuse framework for combining partitionings. In *AAAI 2002*, pages 93–98, 2002.
45. Alexander Strehl and Joydeep Ghosh. Cluster ensembles a knowledge reuse framework for combining partitionings. In *Proceedings of AAAI*, 2002.
46. Y. Teh, M. Beal M. Jordan, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2007.
47. K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *ICML-2001*, pages 577–584, 2001.
48. E.P. Xing, A.Y. Ng, M.I. Jorda, and S. Russel. Distance metric learning with applications to clustering with side information. In *NIPS'03*, volume 16, 2003.
49. T. Xu, Z. Zhang, P.S. Yu, and B. Long. Evolutionary clustering by hierarchical dirichlet process with hidden markov state. In *Proc. IEEE ICDM*, 2008.
50. W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. SIGIR*, pages 267–273, 2003.
51. S. Yu and J. Shi. Multiclass spectral clustering. In *ICCV'03.*, 2003.
52. H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Bi-partite graph partitioning and data clustering. In *ACM CIKM'01*, 2001.
53. H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. *Advances in Neural Information Processing Systems*, 14, 2002.